Miniaturizing, Modifying, and Augmenting Nature's Proteins with Raygun

Kapil Devkota[1,*], Daichi Shonai[2,*], Joey Mao[2], Scott Soderling[2,**], and Rohit Singh[1,2,**]

[1]Dept. of Biostatistics and Bioinformatics, Duke University, Durham, NC, USA

[2]Dept. of Cell Biology, Duke University, Durham, NC, USA

[*]These authors contributed equally to this work.

[**]Co-corresponding authors. Email: {scott.soderling, rohit.singh}@duke.edu

Abstract

Proteins are nature's versatile nanomachines, but engineering them for enhanced function or novel applications remains challenging. Current methods for protein modification struggle to design sequence alterations, especially insertions and deletions, that preserve structure. Here, we introduce Raygun, a template-guided protein design framework that unlocks efficient miniaturization, modification, and augmentation of existing proteins. Using a novel probabilistic encoding of protein sequences constructed from language model embeddings, Raygun is able to generate diverse candidates with deletions, insertions, and substitutions while maintaining core structural elements. We show that Raygun can shrink proteins by 10-25% (sometimes over 50%) while preserving predicted structural integrity and fidelity, introduce extensive sequence diversity while preserving functional sites, and even expand proteins beyond their natural size. In experimental validation, we successfully miniaturize the fluorescent proteins eGFP and mCherry to synthesize functional variants, two of which are smaller than 96% of fluorescent proteins reported in FPbase. Raygun's conceptual innovations in template-based protein design open new avenues for protein engineering, potentially catalyzing the development of more efficient molecular tools and therapeutics.

1 Introduction

Protein design has recently made significant strides, particularly in de novo creation of proteins tailored to specific functions or structures [36, 2, 18, 17, 21]. Yet, the landscape of protein engineering extends

beyond crafting entirely new molecules. A compelling alternative is to build upon existing proteins, an approach we call template-guided design (**Figure 1A-C**). This approach, akin to renovating a building rather than constructing from scratch, has wide-ranging applications such as miniaturizing proteins, modifying established reporters and sensors while maintaining their core functionality, or adapting gene payloads for viral vector delivery. Despite its potential, current template-based approaches face significant limitations. They typically rely on point substitutions– essentially swapping out individual amino acids– and struggle to incorporate more extensive modifications. As substitutions increase, the combinatorial space explodes exponentially (e.g., 25 locations yield 10^{32} possibilities), making computational prediction and experimental validation impractical. Beyond combinatorial substitutions, current methods are fundamentally unable to fully mimic nature's mutational repertoire. Natural evolution generates new proteins not just through substitutions, but also via insertions and deletions (indels). Modifying proteins without the ability to incorporate indels is akin to renovating a building without the ability to add or remove entire rooms. Unfortunately, the field lacks principled, scalable methods capable of accommodating substantial indels while maintaining a protein's core structural form. The potential impact of overcoming this hurdle is profound: a design approach that could manage both combinatorial substitutions and large-scale indels would vastly expand the universe of proteins derivable from a single template.

Protein language models (PLMs) offer a promising avenue for addressing these challenges. By representing proteins as sentences of amino acids, PLMs harness evolutionary conservation to learn distributional rules, generating rich, high-dimensional embeddings of protein sequences: a sequence of n amino acids maps to n points in a high-dimensional space, with each point encapsulating both local and global context of its corresponding residue [22, 25, 19, 6, 13]. PLM embeddings have proven remarkably powerful: they have been applied to predict protein interactions, structures, and functional properties [32, 30, 31, 19, 22, 38]. From a protein design standpoint, the PLM embedding space presents an enticing blend of representational fidelity and computational efficiency. For point substitutions, sampling in the neighborhood of each residue's embedding representation has been shown to generate residue substitutions that preserve peptide binding [4]. We hypothesize that PLMs can be leveraged for template-guided design by reducing the difficult task of sequence-space design to the easier task of embedding-space design. Importantly, sequence–embedding conversions are highly accurate. We use ESM-2, a PLM pre-trained on millions of sequences, for sequence-to-embedding transformation. For the reverse transformation, we trained a small neural network that achieves over 99% validation accuracy; others report similarly high performance [11, 7]. These bidirectional transformations allow us to operate in the structurally-aware embedding space while maintaining a clear path back to realizable sequences, opening new design possibilities. However, the critical challenge of handling indels and combinatorial substitutions remains to be addressed: PLM embedding size varies with sequence length, making proteins of

different sizes incompatibly represented in different-dimensional spaces.

Our key conceptual advance is to encode each protein not as a point in high-dimensional space, but as a probability distribution. To construct this probabilistic encoding, we introduce a fixed-size PLM representation addressing two critical challenges: a) efficient, principled sample generation, and b) bidirectional translation between this representation and variable-length embeddings (and hence sequences). Specifically, we represent any protein of length 50 or higher as a 64,000-dimensional multivariate normal (MVN) distribution. Our design strategy generates candidates by directly sampling from the template protein's MVN distribution. This approach enables "single-shot" design: unlike diffusion-based methods [33, 34, 36, 20] where output quality and diversity depend on intermediate denoising steps, our method generates high-deviation candidates just as efficiently as low-deviation ones, and much faster than diffusion-based methods. We drew inspiration from the central limit theorem (CLT): The CLT states that the sum of identically distributed random variables, given appropriate independence criteria, approaches a normal distribution [12, 28]. We partition proteins of length 50 or greater into a fixed number of segments, each characterized by its own MVN distribution. These segment-level distributions collectively form the full protein's distribution.

We introduce Raygun, a deep learning framework implementing these concepts. An encoder-decoder architecture, Raygun unlocks the generation of diverse, high quality candidate sequences from a given template (**Figure 1**). It offers broad flexibility to the protein designer, allowing them to specify the extent of shrinkage, stretching, and substitutions by two intuitive parameters: the desired output length and the level of noise. The noise parameter broadly controls the substitution rate; indels can be calibrated by changing the output length.

Raygun generates candidates with the desired level of deviation– small or large– from the template while preserving predicted foldability and structure. We observed that sequences of most proteins could be stretched or shrunk by as much as 10% with modest structural impact (median pLDDT [37] decrease against template $\sim$ 17%, TM-score [39] $\sim$ 0.78). At the same time, Raygun can also generate large deviations, such as halving or doubling the protein length or substituting over 50% of the residues, all while broadly preserving the original computed structure. Raygun's speed is valuable during generation: its single-shot generation is 100 fold faster than diffusion-based de novo design approaches and pairing it with PLM-based evolutionary likelihood filters enables us to generate candidates that have large deviations from the template but nonetheless with attractive foldability and structural properties. Across diverse PFAM [3] families, Raygun consistently produces high-quality sequences, preserving PFAM motifs (48.5%) even when substantially altering protein length (50-200% of original length). Unlike some de novo approaches, Raygun's performance is consistent across different secondary structure elements, handling both beta sheets and alpha helices effectively.

As demonstration, we applied Raygun to generate novel fluorescent proteins [29, 16] with properties

not available in any protein currently characterized, natural or artificial. Fluorescent proteins (FPs) are essential tools in cell biology, but their size may disrupt the functions of proteins they're fused to, especially small proteins. Using Raygun, we generated 8 miniaturized candidates across two FP templates. Five of these candidates showed fluorescent activity matching the spectrum of the original template, albeit at lower intensity. Notably, two of the generated proteins, at lengths of 199 and 206 amino acids, both derived from mCherry, are shorter than 96% of fluorescent proteins reported in FPbase [16]. One of these deviated from the chromophore sequence of its template (mCherry), underscoring Raygun's potential to synthesize new-to-nature designs.

Raygun's high hit-rate, both computationally and experimentally, combined with its speed and tunability, opens new avenues in protein design and engineering. Conceptually, Raygun represents a novel application of protein language models, utilizing a rich fixed-size representation that extends beyond discriminative learning to enable generative tasks. This approach has the potential to accelerate protein engineering across a wide range of applications, from developing more efficient molecular tools to designing novel therapeutics.

2 Results

2.1 A fixed-sized language for all proteins

Constructing a model that accepts variable-length PLM embeddings and generates viable candidate sequences of any length presents significant design challenges. Our objective of allowing the user any choice of output sequence length necessitates a fully length-agnostic protein representation at some internal stage. The generative model must robustly sample in this length-agnostic space, and subsequent stages must map the fixed representation back to variable lengths accurately. To address these constraints, we designed Raygun as an autoencoder. The encoder performs the variable-to-fixed transformation, while the decoder reverts the fixed-length embedding to the variable length space.

Ensuring effective variable-to-fixed mapping in the Raygun encoder was crucial to the model's success. ESM-2 embeddings, trained via masked language modeling (MLM), provide local and global protein information at each residue. We hypothesized that these variable-length embeddings have enough redundancy to allow fixed-length compression with minimal information loss. To address this, we introduced a two-level representation. First, we modeled any variable-length ESM-2 embedding as a series of K contiguous blocks, focusing on template proteins of at least 50 residues. We chose $K = 50$ after a preliminary comparison between $K = 50$ and $K = 25$ (**Figure A.3**). We posited that the series of K blocks could be trained to recapitulate the template's global representation. Each block's size depends on the template protein length (e.g., 10 for a 500-length template), but we envisioned that the per-residue embeddings have enough redundancy to be reduced to a fixed-size representation without significant information loss.

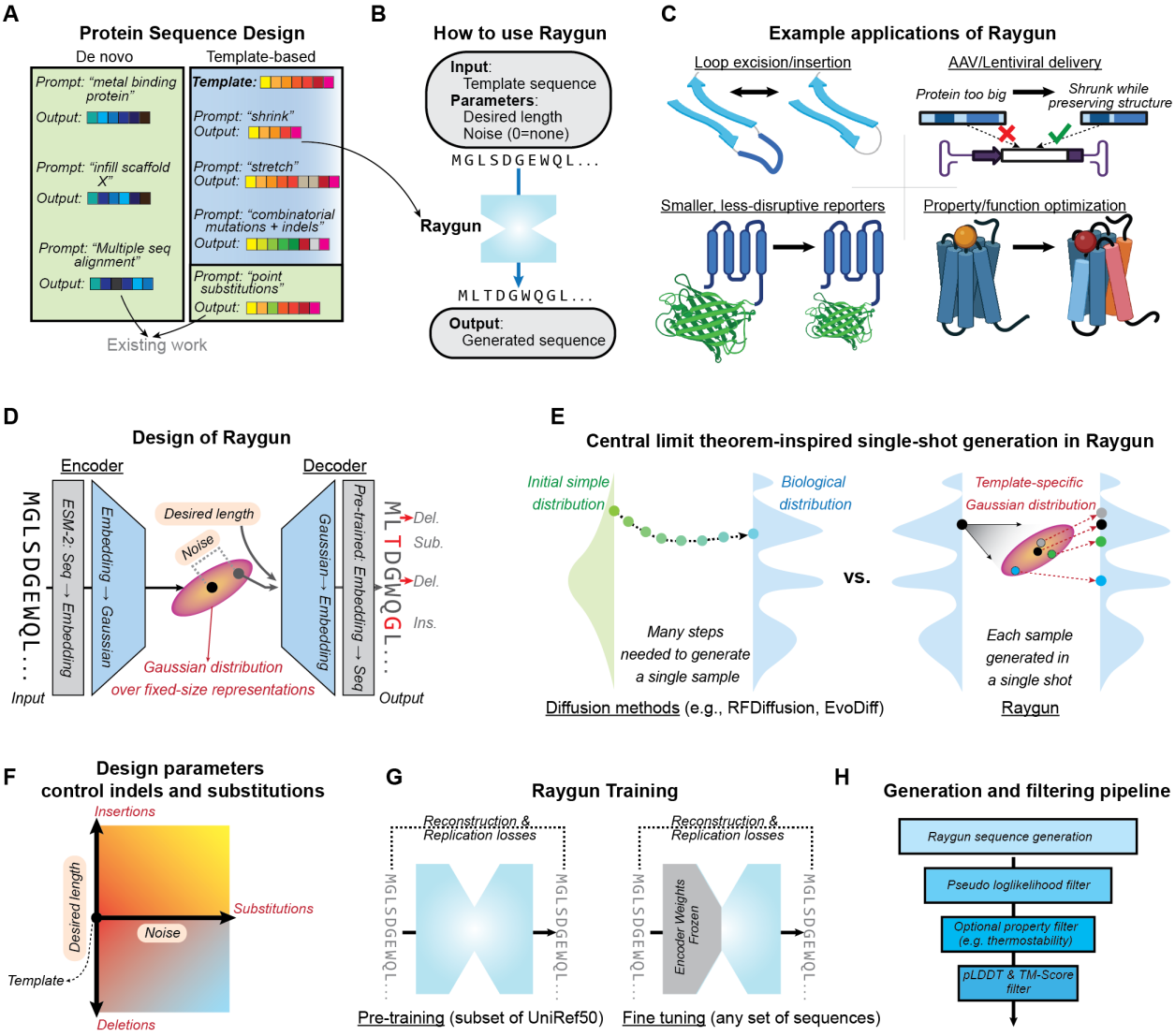

Figure 1: Description of the Raygun Model. **(A)** Unlike de novo models, which design entirely novel proteins conditioned on structural or functional constraints, Raygun uses existing proteins as a template and introduces appropriate substitutions and indels in a template-guided design. **(B)** Raygun takes only three inputs: the template sequence, a noise parameter and target length. **(C)** Raygun has direct and wide-ranging applications in biological research, ranging from protein sensor design to protein miniaturization for gene therapy. **(D, E)** Raygun works as an auto-encoder model. The encoder converts variable-length sequence input into a fixed-dimensional representation, and this space approximates a multi-variate normal distribution. Unlike diffusion based models, single-shot sampling can be done through this distribution. **(F)** Raygun parameters provide fine-grained control over indels and substitutions. **(G)** Raygun is trained in a self-supervised way. For inference, we recommend using a fine-tuned version in which only the decoder block is updated. **(H)** Raygun is fast (0.3 seconds per generation) and can be paired with downstream filtering steps to generate high quality candidates efficiently.

Averaging across a block's length should, by the Central Limit Theorem for weakly dependent variables, approximate a Multivariate Normal Distribution (MVN), which can be sampled directly. Thus, the Raygun encoder can produce a fixed-length embedding space that is tractable for sampling and retains

sufficient information for the Raygun decoder to accurately map it back to variable lengths.

2.2 Raygun architecture & training

The Raygun architecture consists of two length-transforming layers, "Reduction" and "Repetition," and multiple length-preserving "T-Block" layers. The Reduction and Repetition layers are parameter-free and operate in the encoder and decoder stages, respectively. T-Block layers, present in both stages, are deep neural network modules trained to optimize embeddings by enhancing latent local and global properties. Each T-Block layer comprises an ESM Transformer for global properties and a 1D-convolution block for local relationships. They are used before, between, and after the length-transforming layers (**Methods A.1**). T-Block parameters comprise most of Raygun's 701 million trainable parameters.

In each Reduction layer, within-block averaging generates two fixed-length outputs per block: the block-wide mean and standard deviation matrices, describing the MVN distribution of the fixed-length space. During inference, the user-supplied noise scale dictates the sample generated from this MVN for subsequent layers; during training, the MVN's mean is used. The Repetition layer accepts a target length and a fixed-length representation from the encoder stage, producing a variable-length embedding of the desired length.

Training the Raygun model We formulated Raygun training as a self-supervised problem. Specifically, the model is trained to compress the input sequence to a fixed size, then decompress it back to its original length with maximal fidelity. We assessed the quality of the reconstructed output in both the embedding and sequence spaces, by using two losses: (a) reconstruction loss, penalizing deviations in the reconstructed embeddings, and (b) cross-entropy loss, penalizing deviations in the sequence space after using a pre-trained ESM-2-decoder. Additionally, we sought self-consistency in the fixed-length embedding. For each input sequence, we decoded the fixed-length representation to a shorter-length sequence, re-encoded it, and compared the two fixed-length representations under a replication loss (**Methods A.2**). Together, these three losses comprised the overall training objective.

We paid special attention to curating the training data, reasoning that good generalizability would benefit from a wide, even distribution of protein lengths. After inspecting the length distribution in Uniref-50, we focused on proteins of lengths between 100 and 1000, dividing this range into 19 bins. For each bin, we randomly selected roughly 5000 sequences within the specified length range from the Uniref-50 database, resulting in a total of 94,734 proteins. Finally, of these extracted Uniref-50 proteins, we randomly chose 80,000 proteins for model training and used the remaining 14,734 proteins for validation and hyper-parameter selection. We trained the model for 15 iterations; in validation set, the largest Blosum-weighted [9] sequence identity score (or simply "Blosum score") of 0.52 was observed at epoch 12 (**Method A.2**). We therefore chose this epoch-12 model as our pre-trained model for further experiments.

An alternative to this self-supervised approach would be supervised training where pairs of sequences

with differing lengths but identical structural and functional properties are used. Unfortunately, this is an infeasible approach. We could not find experimental datasets where large-scale indels were performed with this goal in mind. Databases like SCOP [23] and CATH [24], while grouping proteins by structural domains, are domain-focused rather than ensuring structural and functional similarity at the whole-protein level.

Raygun fine-tuning and sampling Akin to pre-trained foundation models that are fine-tuned for specific use-cases, we recommend fine-tuning Raygun before a protein design campaign. This can be done simply and efficiently. For a single protein, or a group of proteins (e.g., the relevant PFAM domains), a simple FASTA file can be provided as input to fine-tune the autoencoder; no alignment is needed and diverse sequences can be included. The fine-tuning is designed to improve the fidelity of Raygun decoder for the particular protein group, so that the decoder can accurately transform the fixed-length representation back to the variable length sequence space with the Blosum score of > 0.99. To avoid the risk of catastrophic forgetting seen sometimes in fine-tuning, and noting that the actual sampling is controlled by the encoder segment of Raygun, we froze the encoder weights while doing fine-tuning optimization; only the decoder is fine-tuned to ensure the generation is well-matched to the proteins of interest.

Although Raygun can perform single-shot protein generation (unlike other diffusion based models), we found that using a one-step recycling iteration improves performance. Similar to the diffusion process, where an intermediate output is again passed to the input, recycling in Raygun takes the generated candidate and uses it as a template to generate a new candidate. We noted that it improves the quality and diversity of output samples (**Method A.3.5**).

Our generation process is fast enough (0.3 secs/iteration on an A100 GPU) that, even after one-step recycling, we can generate a large number of viable candidates for moderately sized proteins of length $>$ 1000. Separately, we note that Raygun works well even on very large proteins (like mTOR), albeit slower. Raygun's speed gives us the flexibility to apply filtering schemes to select higher-quality candidates from the generated samples. We primarily used a PLM-based metric called pseudo-log Likelihood (pLL), which assesses the fitness of a protein as modeled by language models like ESM-2 [5]; we devised a length-adjusted version (**Method A.3.1**). For each generated sample, we employed pLL to rank the candidates by their scores, selecting those with the highest fitness. While other PLM and structure-based filtering techniques were used in some experiments, pLL was consistently the first filtering step due to its computational efficiency.

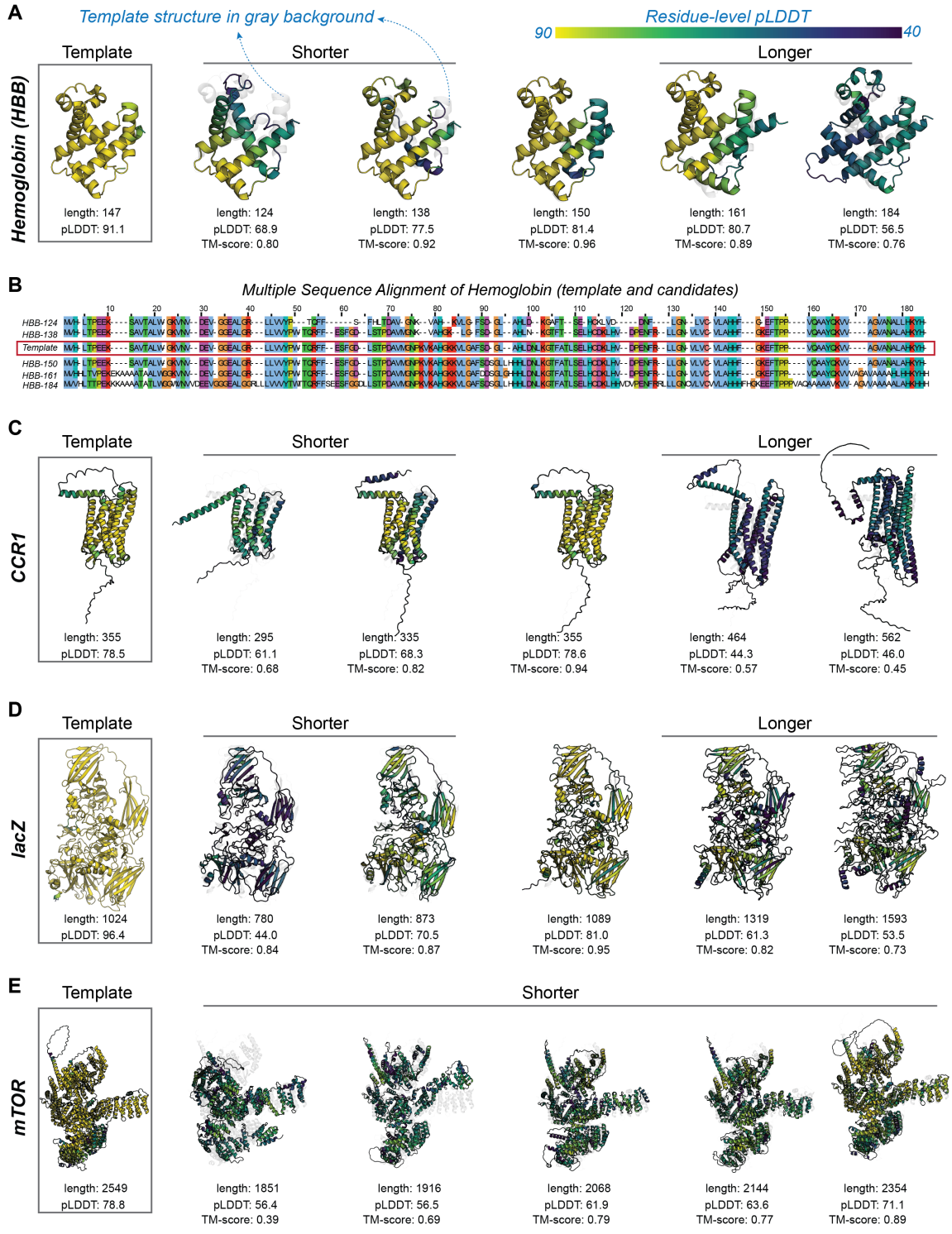

Figure 2: Protein editing using Raygun for proteins of different sequence lengths. The following proteins were chosen for the demonstration of Raygun's shrinking/enlarging capabilities: **(A)** Hemoglobin (HBB), **(C)** CCR1, **(D)** lacZ and **(E)** mTOR. For HBB, the multiple sequence alignment of the Raygun candidates and the templates are shown in **(B)**. For HBB, CCR1 and lacZ, we showed both the shrunk and enlarged results. For mTOR, however, owing to its large size, we show only its selected shrunk candidates.

2.3 Raygun preserves protein structure while introducing indels & combinatorial mutations

Raygun effectively generates novel candidates across a wide range of template sizes while largely maintaining the candidates' structure and integrity. As a demonstration of this capability, we applied Raygun to a diverse set of proteins (**Figure 2**), including hemoglobin (147 aa), CCR1 (355 aa), lacZ (or BGAL_ECOLX; 1029 aa), and mTOR (2549 aa). Our goal was to generate both shortened and elongated, variants of these proteins, assessing Raygun's ability to introduce insertions, deletions, and substitutions while preserving overall computed structure.

We generated 2000 samples for each protein across a broad range of lengths, setting Raygun's sampling noise parameter to 0.5 to introduce moderate sequence variations. To ensure the quality of our candidates, we filtered the samples using length-adjusted ESM-2 pseudo-loglikelihood (pLL) [5] scores, which have been demonstrated to correlate with evolutionary fitness of a sequence. We retained the top 5% of generated sequences distributed across the length spectrum (**Methods A.3.1**).

Figure 2 shows the AlphaFold3-predicted structures of five representative candidates for each protein, along with their predicted Local Distance Difference Test (pLDDT) scores and alignment with the template (TM-scores). The results demonstrate that Raygun effectively preserves protein structure across a wide range of output lengths, as evidenced by the high TM-scores when comparing generated structures to their respective templates.

Raygun's performance extends from small proteins to large, complex ones like mTOR (2549 aa). However, we observed that the degree of structure preservation varies depending on the properties of the template protein. For instance, among the samples with inferred AlphaFold3 [1] structures, CCR1's TM-score decreased to 0.68 when shortened by 17%, while mTOR could accommodate a 25% reduction to reach a similar TM-score of 0.69. As expected, increasing deviations from the original length generally correspond to gradual decreases in TM-score and pLDDT values. This trend reflects the challenge of maintaining structural integrity as proteins undergo more substantial modifications. While we used a moderate noise-factor of 0.5 for these samples, we will shortly show how variations in this parameter control sequence and predicted structural similarities of the generated proteins.

Desiderata for effective template-guided protein design To establish a comprehensive framework for evaluating template-guided protein design methods, we propose a set of critical capabilities that such tools should possess. These criteria not only guided our assessment of Raygun but we expect will also be useful for future template-based design approaches.

- Structural versatility: An ideal method should generate diverse sequences without bias towards specific secondary structures. This is crucial, as many de novo methods favor α-helices over other structural elements [2].

- Functional site preservation: Maintaining critical functional regions is important in protein engineering. We assessed Raygun's capacity to retain known active and binding sites of template proteins in the generated samples, ensuring that key functional properties are preserved.
- Tunable modification range and sequence diversity: A versatile design tool should enable both minor tweaks and major alterations to the template, introducing indels and substitutions to produce a spectrum of variants with high sequence diversity while maintaining essential structural characteristics.
- Scalability across protein sizes: An effective method should perform consistently across a range of protein sizes. This criterion ensures broad applicability in diverse protein engineering scenarios.
- Information-rich representation: The quality of the fixed-length representation used in the design process is crucial. It should encapsulate sufficient structural information to guide accurate sampling. We evaluated this by measuring the representation's ability to cluster structurally related sequences and compared it to baseline ESM-2 embeddings.

In assessing Raygun's performance on these criteria, we have sometimes focused our analyses below on Raygun-generated candidates that were shortened relative to their templates. This approach allowed for straightforward sequence alignment-based evaluation metrics. We posit that these assessments will generalize to expanded candidates, given that Raygun employs the same underlying mechanism for both shrinking and expanding proteins.

Raygun miniaturizes proteins with a slight preference for deleting less-structured regions To evaluate Raygun's structural versatility, we analyzed its behavior when miniaturizing proteins across various secondary structure elements. From the PDB database, we selected 10,000 template proteins belonging to SCOP [23] families representing α, β, $\alpha + \beta$, and α/β structural classes. For each template, we generated shortened variants using Raygun and performed pairwise alignments between the miniaturized candidates and their original templates.

We categorized secondary structure elements (SSEs) into three groups: α helices, β sheets, and loops. Here, "loops" encompass all regions not annotated in the PDB file as α helices or β sheets, including intrinsically disordered regions. By examining the gaps in these alignments, we could infer which structural elements Raygun preferentially removes during the miniaturization process (**Figure 3A**).

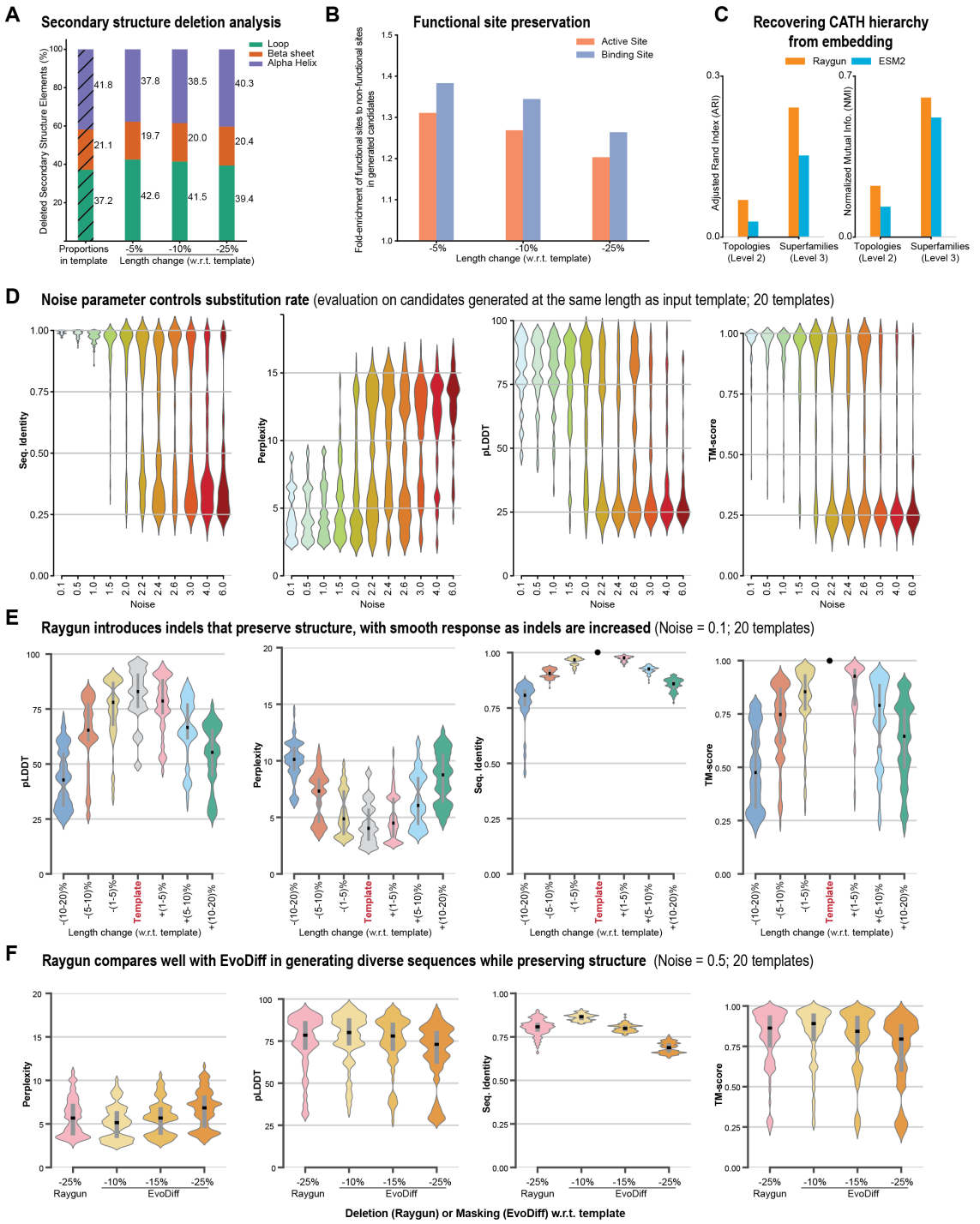

Figure 3: Evaluation of Raygun's template-guided design: **(A)** Deletion preference of Raygun for secondary structure elements (SSEs). Here, "loops" refers to all non α or β residues. Raygun operates equitably across SSEs, with a slight preference for deleting loops. (*w.r.t.* = with respect to). **(B)** Ratio of Raygun preserved functional sites to the overall sequence preservation, during deletion. The ratio is consistently greater than 1, indicating that Raygun preferentially preserves active and binding sites **(C)** Clustering comparison between ESM-2 and Raygun's fixed-length embeddings. The ARI and NMI scores show that the fixed-length embeddings produce clusters that are better aligned with the CATH structural hierarchy. **(D, E)** Structural and sequence measures for Raygun candidates at different noise and length-modification regimes. (D) The noise parameter controls substitution rates. (E) Proteins can be stretched/shrunk by upto 10% with modest loss in predicted structural fidelity. **(F)** Same-length comparisons with EvoDiff, where a part of the sequence has been masked and the methods are tasked with regenerating them. At the same masking rate as EvoDiff, Raygun's candidates have better predicted structural properties.

Our analysis reveals that Raygun is broadly equitable in removing various SSEs, with a slight preference for removing less-structured regions. For candidates shortened by 5%, 42.6% of the gaps in the pairwise alignment corresponded to loops, although loops comprised only 37.2% of all secondary structures in the original templates. This indicates a modest bias of approximately 5% towards removing loop regions. As the degree of miniaturization increases, this preference becomes muted, leading to an even greater balance in deletions. At 25% length reduction, loops account for 39.4% of removed regions, only a 2.2% increase over the original templates. The relative propensity to remove α helices vs. β sheets remains broadly unchanged.

These results demonstrate Raygun's structural versatility, showing that it can effectively miniaturize proteins while maintaining a balanced approach to different secondary structure elements. Its slight preference for deleting loops aligns with the general understanding that such regions often contain fewer structurally important elements compared to α helices and β sheets. However, it's important to note that loops can still play crucial functional roles in proteins; we explore this next.

Raygun preferentially preserves active and binding sites during miniaturization We analyzed 10,000 proteins from UniProt with annotated active and binding sites to assess Raygun's ability to preserve functionally important regions during miniaturization. For each protein, we generated candidates at 5%, 10%, and 25% length reductions and performed pairwise alignments with their templates.

We compared the preservation of active and binding sites to overall sequence conservation. Specifically, we calculated the ratio of preserved functional sites to the overall sequence identity between the miniaturized candidate and its template. A ratio greater than 1 indicates that functional sites are conserved more than would be expected by chance. Figure 3B shows that this ratio consistently exceeds 1 across all deletion settings, with the highest preservation observed at 5% length reduction. Binding sites are retained slightly more than active sites. These results demonstrate that Raygun effectively captures and prioritizes functionally important regions during the miniaturization process, even without explicit annotation of these sites in its training data. As expected, preservation decreases with more extensive deletions, underscoring the challenge of maintaining functional integrity during major structural modifications.

Raygun parameters provide fine-grained control over protein generation Raygun offers precise control over protein generation through two key parameters: noise and output length. The noise parameter scales the covariance matrix of the multivariate normal distribution from which we sample, allowing fine-tuned control over sequence variability. To evaluate the impact of these parameters, we

conducted experiments on 20 proteins spanning different structural classes (these are the same template proteins as used in the PFAM analysis in Figure 4).

First, we assessed the effect of noise on sequence and structural properties while maintaining the template length (**Figure 3D**). For each template, we generated samples for noise values ranging from 0.01 to 6, producing 100 candidates per setting and retaining the top 5 after filtering. As noise increases, sequence identity gradually decreases until an inflection point around 2.2, after which the decline becomes steeper. Structural metrics like TM-score (against template) and pLDDT follow similar trends. These results suggest using noise values near 0 for minor edits and approaching 2 for greater sequence variability.

Next, we examined how changes in output length affect protein properties at three noise levels: 0.1 (**Figure 3E**), 0.5 and 1.0 (**Figures A.8,A.9**). Raygun introduces short or long indels as needed to achieve the desired length while balancing structural preservation. As expected, larger deviations from the template length lead to decreased sequence identity and TM-score. However, this degradation is more gradual compared to noise-induced changes: Raygun largely maintains structural similarity for length changes within $\pm 10\%$ of the template. In specific cases, even larger edits can broadly preserve the structure; e.g., mTOR could be miniaturized by over 500 residues ($> 20\%$ shrinkage) with a TM-score of approximately 0.7 (**Figure 2E**).

These experiments demonstrate that Raygun's parameters offer a flexible toolkit for protein design. The noise parameter primarily controls substitution rate, while output length determines the extent of indels. This fine-grained control allows for a spectrum of designs, from subtle tweaks to major structural modifications, with predictable trade-offs between sequence diversity and structural preservation.

Raygun compares favorably with existing approaches in balancing sequence diversity and structural plausibility Raygun's novel approach to protein design, by introducing substantial indels and mutations in templates, presents a challenge for direct comparisons with existing de novo design methods. To establish a meaningful benchmark, we developed a comparison framework based on sequence masking and regeneration—- a task that allows us to compare Raygun with EvoDiff, a sequence-based de novo design method. We mask a portion of the input sequence and task both methods with regenerating the masked regions. This approach allows us to evaluate how well each method balances sequence diversity and structural plausibility when faced with incomplete information. We applied this framework to our set of 20 diverse proteins; for Raygun, we set the noise parameter to 0.5, allowing moderate sequence edits. **Figure 3F** illustrates the results of this comparison.

At 25% masking for both Raygun and EvoDiff, the former improves over the latter across all structural metrics. Since masking impact may not be directly comparable across the two methods, we also evaluated

EvoDiff with 10% and 5% masking. At these lower masking rates, EvoDiff's perplexity, pLDDT, and TM-score become comparable to Raygun's results at 25% masking. However, EvoDiff's sequence identity with the original increases significantly at these lower masking rates, indicating less diverse sequences. These results highlight Raygun's ability to maintain a favorable balance between sequence diversity and structural plausibility. For a given level of structural quality (as measured by pLDDT or TM-score), Raygun generates more diverse sequences. While we do not anticipate this to be the primary way Raygun will be used, this analysis demonstrates Raygun compares favorably with state-of-the-art de novo protein design techniques in its ability to reach new parts of the protein space.

Raygun's fixed-length representations better capture the organization of protein structural shapes PLMs have demonstrated a remarkable ability to capture fine-grained structural similarities between proteins: at short ranges in embedding space, proteins with highly similar structures cluster together. However, for effective protein design it is crucial that the embedding space also captures meaningful relationships at larger distances. A well-organized representation that captures both fine and coarse structural similarities could enable smoother transitions between related structures, facilitating tasks such as protein miniaturization or expansion. Towards this, we investigated whether Raygun's fixed-length representations improve upon ESM-2 embeddings in capturing these multi-scale structural relationships, particularly at coarser levels of protein structure classification. We leveraged the CATH database, which classifies proteins into hierarchical structural categories, to evaluate the clustering performance of Raygun and ESM-2 representations. Our analysis focused on the three main top-level classes of the CATH hierarchy: "mainly alpha", "mainly beta", "alpha beta". Within these, we focused on the two top levels of the hierarchy: architecture (e.g., "Up-Down Bundle"), and topology (e.g., "Bromodomain-like").

For architecture and topology levels of CATH, we performed agglomerative clustering using Raygun or ESM-2 average-pooled embeddings. We filtered the dataset for 60% sequence identity and only chose CATH groupings containing at least 50 sequences (Methods). **Figure 3C** shows the clustering performance measured by Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). Both Raygun and ESM-2 representations performed better at distinguishing finer-grained topologies compared to coarser architectures. However, Raygun consistently outperformed ESM-2, with the gap being particularly pronounced at the coarser architecture level.

These results suggest that Raygun's fixed-length representations not only retain but potentially refine the structural information present in ESM-2 embeddings. The superior performance at the coarser level indicates that Raygun captures a better hierarchical organization of protein structure space. This

property could be particularly valuable for protein design tasks where preserving overall topology is crucial, even while when introducing significant sequence alterations.

Raygun demonstrates structural versatility while preserving functional domains To further assess Raygun's ability to generate diverse structures at various lengths while maintaining functional integrity, we applied it to preserve PFAM domains across a wide range of structural classes, and input and output protein lengths. We selected PFAM domains corresponding to the four major SCOP classes to ensure structural diversity: α-only (7 transmembrane receptor family of GPCR proteins, PF00001), β-only (a family of beta-propellers, PF10282), α/β (YdjC-like proteins, PF04794), and $\alpha + \beta$ (F420 ligase family, PF01996).

For each domain, we chose five templates of diverse lengths and, from each template, generated candidates ranging from 50% to 200% of the median template length in the domain. Our sampling strategy divided this length range into 20 buckets. For each template and length bucket, we initially generated 100 Raygun candidates, resulting in 10,000 ($=5 \times 20 \times 100$) candidates per domain. We then applied a filtering step based on pLL scores, retaining the top 5 candidates per bucket and template combination, yielding 500 filtered candidates per PFAM domain. To estimate function preservation, we used HMMER to identify candidates that retained their corresponding PFAM domain.

The results demonstrate Raygun's robustness across structural classes (**Figure 4**). It was able to generate structurally diverse candidates across a broad range of lengths while maintaining functional domains. On average, 48.25% of candidates retained their PFAM domains across all four structural classes. The retention rates ranged from 37% for the most challenging $\alpha+\beta$ class to 57% for the α/β class, demonstrating Raygun's robustness even when dealing with complex protein folds. This structural versatility, combined with functional preservation, positions Raygun as a powerful tool for protein engineering tasks that require significant alterations while maintaining the template's core structural/functional properties.

2.4 Experimental analysis of Raygun 's ability to miniaturize proteins while retaining function: generating novel short fluorescent proteins

We next experimentally tested the ability of Raygun to shrink a class of proteins that have revolutionized cell biology: fluorescent proteins (FPs) [29], used for over 60 years, have transformed live-cell imaging by enabling the direct visualization of protein localization and dynamics in situ. Extensive engineering of FPs has produced variants with distinct chromatic properties, monomeric forms, maturation times, and stability, making them indispensable tools for a wide range of applications [8, 26]. However, less attention has been given to reducing their size, a critical consideration for studies involving small proteins, which

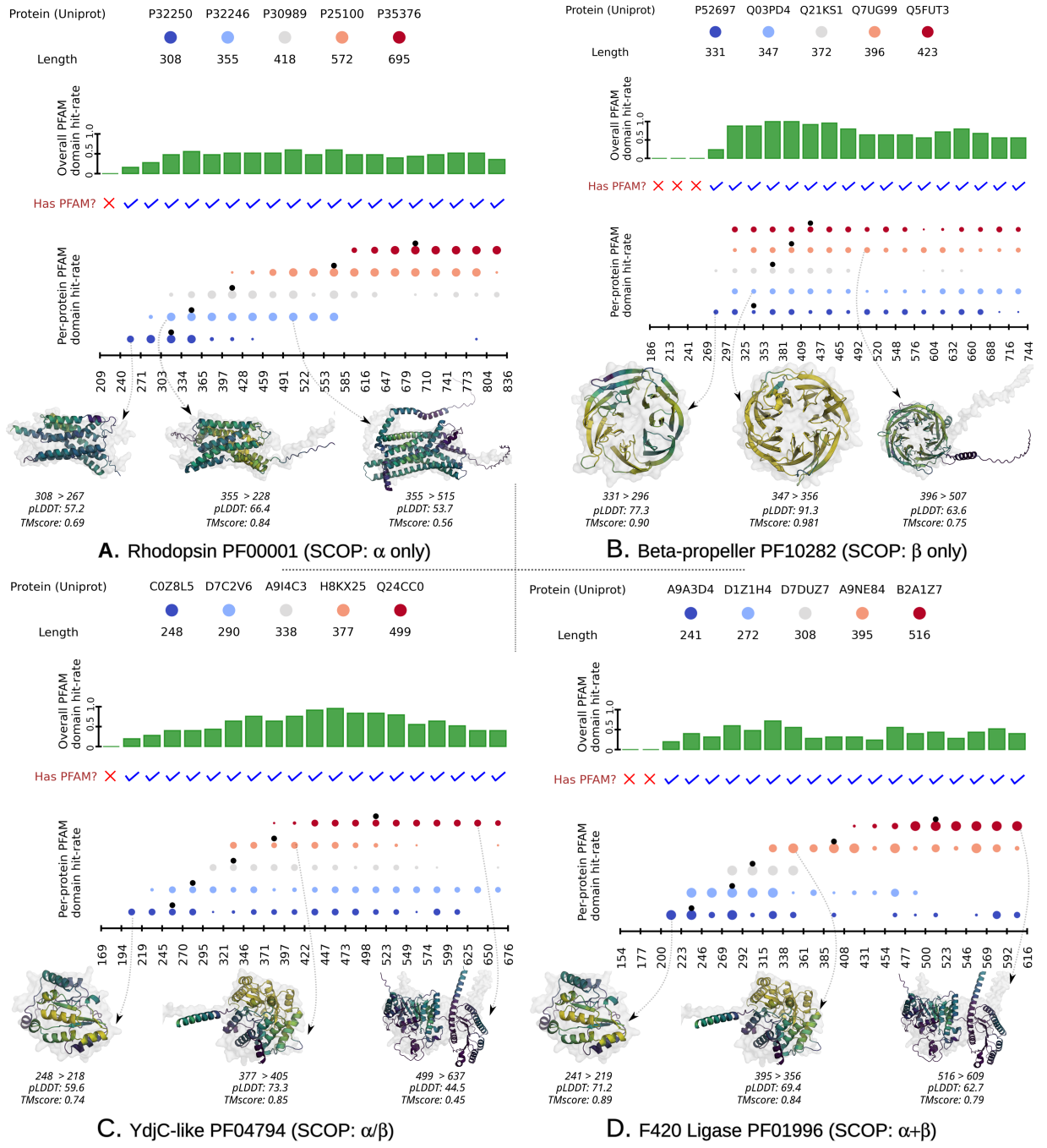

Figure 4: Demonstrating Raygun's effectiveness in preserving diverse structural features, as indicated by PFAM domains. We evaluated across all four major SCOP classes: **(A)** α (*PF00001*), **(B)** β (*PF10282*), **(C)** α/β (*PF04794*), **(D)** $\alpha+\beta$ (*PF01996*). For each PFAM domain, we obtained 5 representative proteins of diverse sizes and used Raygun to generate 100 samples (per template) spanning a wide length interval (50-200% of median protein length). After dividing the overall interval into 20 uniform length-bins, we report the number of Raygun candidates with retained PFAM domains across each bin. Additionally, for a selected number of candidates in each domain, we also provide their AlphaFold3 inferred structures and metrics (pLDDT and TM-score). The template protein's structure is shown as gray background.

may be disrupted by the relatively large FP fusions. Given the central role of FPs in cellular research, and the ongoing efforts to optimize their performance, we tested Raygun's ability to reduce the lengths of these proteins while preserving their essential function, i.e., fluorescence.

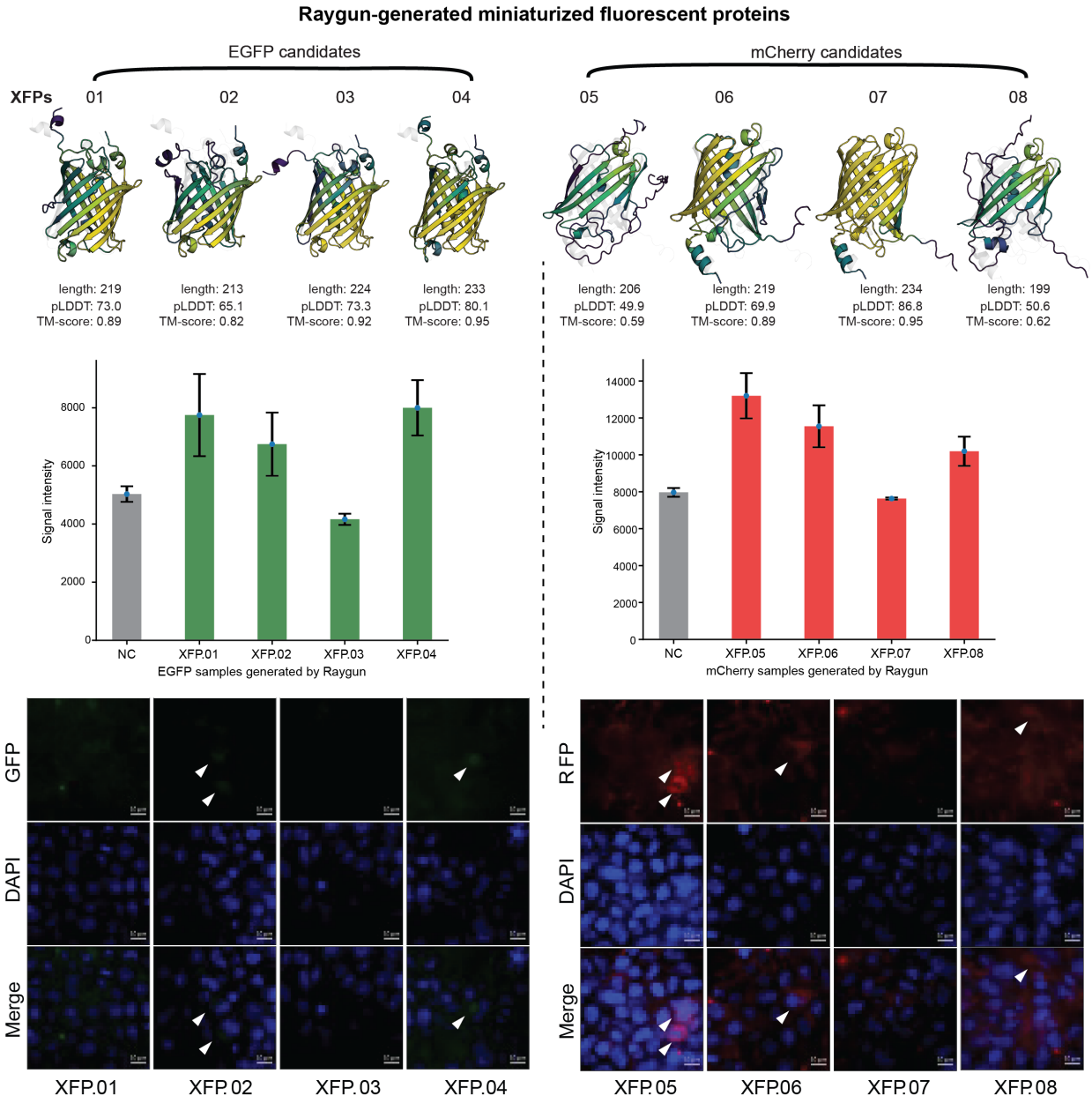

Figure 5: 8 Raygun generated XFP candidates selected for biological evaluation. The AlphaFold3 predicted structure and metrics(pLDDT and TM-score against the template) for each candidate is provided in the first row. The fluorescent images and the corresponding signal intensities for the selected candidates are provided in the succeeding rows. We observed that XFPs 02, 04, 05, 06 and 08 showed significant fluorescence over negative controls, while the XFPs 03 and 07 were confirmed to lack activity.

We used Raygun to downsize two widely-used fluorescent proteins: eGFP (238 aa), a classic green fluorescent protein, and mCherry (236 aa), a red-emitting fluorescent protein. For each template, Raygun

generated four candidates with lengths ranging between 195 and 235 amino acids. These candidates were selected after filtering 70,000 Raygun-generated samples for each template. The filtering process began with a pseudo log-likelihood (pLL) filtering, which eliminated 90% of the samples. Next, we applied an HMM-based tool called "hmmscan" to remove samples lacking the correct PFAM domain. Finally, we used a custom brightness prediction tool, trained on the GFP brightness dataset [27, 16], to discard samples with low brightness. The detailed sampling procedure is provided in the **Methods A.3.3, A.3.4**. This filtering pipeline resulted in the generation of 8 FPs, XFP01-04 and XFP05-08 being the eGFP and mCherry candidates respectively (**Figure 5**).

Our filtering process did not explicitly select candidates based on the presence of the characteristic chromophore sequence, an essential 3 amino acid sequence in the form of XYG typically located at positions 65 to 67 in GFP. Recently, Hayes et al. [13] took a scaffolding-based approach to fluorescent protein, including specifying the chromophore. We did not provide any such information to the model: we merely specified the noise and length criteria and let Raygun generate miniaturized FP candidates.

We then proceeded to experimentally test fluorescent properties of XFP01-08. For each candidate, codon-optimized cDNAs were synthesized, cloned into expression vectors and transfected into HEK293 cells. Four days later images were taken and manually inspected for fluorescence over background. Six XFPs appeared to display fluorescence over negative control and were selected for further analysis by quantitative image analysis along with two variants that did not appear to have activity. Analysis revealed significant fluorescence over negative control for five of the selected Raygun variants (XFPs 02, 04, 05, 06 and 08), while XFPs 03 and 07 were confirmed to lack activity. Importantly, Raygun was able generate functional FPs from diverse evolutionary sources, eGFP originating from the jellyfish *Aequorea victoria* while mCherry originates from the coral *Discosoma*. The evolutionary distance between the two groups, Hydrozoa (jellyfish) and Anthozoa (corals), is estimated to be around 600 million years and each FP has different properties, green versus red fluorescence. Despite this Raygun effectively shortened each by up to 25 aa (10.5%) for eGFP and 37 aa (15.6%) for mCherry.

3 Discussion

Raygun introduces a novel approach to template-guided protein design, leveraging probabilistic encoding and language model embeddings to enable extensive modifications while preserving core structural elements. This method allows for both minor tweaks and major alterations, enabling a spectrum of protein variants with high sequence diversity. The capabilities demonstrated by Raygun have far-reaching implications for protein engineering. For instance, we generated functional fluorescent proteins (FPs)

with lengths of 199 and 206 amino acids, shorter than 96% of FPs listed in FPbase. This is particularly significant given the ubiquity of FPs in cellular research and the ongoing efforts to optimize their performance.

Raygun's core innovation lies in representing proteins as probability distributions in a fixed-size embedding space. This approach addresses the challenge of handling variable-length sequences in protein design. By encoding proteins as multivariate normal distributions with fixed parameters, Raygun enables efficient sampling and manipulation of sequences regardless of length. This probabilistic encoding, combined with rich protein language model embeddings, creates a flexible framework for generating diverse protein variants while maintaining core structural properties.

Raygun's template-based design approach sets it apart from de novo approaches that have been the focus of recent protein design efforts. Template-guided approaches like Raygun offer distinct advantages, particularly in scenarios where modifying existing proteins is preferable due to immunogenicity concerns, compatibility with existing workflows, or the need to minimize in vivo disruption. Moreover, there are certain applications (e.g., miniaturization, or insertion of specific binding, localization, or signaling motifs) where template-guided design is the only feasible approach.

Our experimental validation of FPs suggests that template-guided approaches can, in some cases, enable greater novelty than de novo design. Consider our FP generation approach against a recent de novo GFP design by Hayes et al. Using a new PLM, ESM-3, the latter relied heavily on scaffolding constraints when generating 'esmGFP': their approach preserved the length (229) to be the same as the PDB exemplar 1QY3, specified not only the sequence but also the structure of residues critical for chromophore formation (Thr62, Thr65, Tyr66, Gly67, Arg96, Glu222), as well as the structure of residues 58-71, deemed crucial for chromophore energetics. In contrast, we did not impose any such constraints. Notably, most of Raygun's generated candidates did preserve the chromophore and, given Raygun's speed, ensuring this for all candidates by explicit filtering would have been straightforward. However, we sought to explore if other chromophores could also work, as this would enable generation of potentially new functionality not available in nature. Therefore, in choosing which candidates to experimentally validate, we included a few with non-canonical chromophores. In particular, the 199-length FP we generated and validated has glycine deleted, its position taken by a serine instead. This demonstrates Raygun's capacity to explore a broader design space, potentially uncovering novel protein variants that more constrained approaches might miss. In any protein design strategy, there is a balance between staying close to the manifold of feasible proteins while being distinct from the subset of natural proteins. De novo design methods start far from the manifold and try to end at a candidate that resides on it. Raygun, in contrast, starts from the manifold and seeks to stay on it while moving away from the

starting point. We speculate that the latter approach may actually provide a better conceptual balance in reaching novel but feasible proteins.

Our PLM-based approach is very efficient-— its single-shot generation takes about 0.3 seconds per iteration on an NVIDIA A100 GPU, about 100-fold faster than the de novo method EvoDiff, which itself is faster than other de novo design methods. This speed, combined with the use of PLM-based evolutionary likelihood filters, enables rapid generation and screening of candidates that deviate significantly from the template while maintaining attractive foldability and structural properties. For instance, in our fluorescent protein experiments, we were able to generate and filter 70,000 samples for each template, applying multiple layers of computational screening before experimental testing. This high-throughput computational approach, coupled with a high experimental hit-rate (5 out of 8 tested FP variants showed significant fluorescence), demonstrates Raygun's potential to accelerate the protein engineering pipelines. Here, we achieved our results using a relatively modest PLM (ESM-2 650M) and a training set of only ∼95,000 sequences. There is significant potential for improvement by incorporating more powerful PLMs, larger training sets, or structure-infused models. The recent advances in PLMs, such as ESM-3 [13] and SaProt [35], suggest that integrating these more sophisticated models could further enhance Raygun's performance.

Our work highlights several underappreciated aspects in the field of protein design. First, standard computational metrics for assessing generation quality, such as inverse folding, perplexity, and pLDDT scores, may not always correlate with in vivo or in vitro functionality. For instance, our smallest mCherry variant exhibited poor pLDDT scores and a different chromophore sequence, yet demonstrated fluorescence on par with other positive hits. This underscores the importance of experimental validation and the potential limitations of relying solely on computational metrics.

Secondly, our work suggests that the relationship between sequence and function is more complex than might be currently appreciated, especially for highly engineered proteins like FPs. Both we and Hayes et al. found that the generated FP candidates are dim and will require directed evolution to reach higher fluorescence. This speaks to the discussion in the field about the kind and amount of experimental training data needed for computational protein design. Towards a low-data approach, Hie et al. have hypothesized [14] that PLM-generated mutations, which have been shown to follow evolutionary rules, are likely to improve fitness in general; they demonstrated this in the case of particular antibodies. This is an appealing hypothesis. If true, it could obviate the need for function-specific experimental data to guide protein optimization. Unfortunately, while this hypothesis may hold true for functions that are evolutionarily rooted, our work with fluorescent proteins suggests that for engineered or non-natural protein functions, the fitness landscape may be more rugged and discontinuous than this hypothesis

implies. While the evolutionary insights captured by PLMs are very valuable in preserving structure, that may not always be sufficient to guide the optimization of non-natural or highly specialized protein functions. This suggests that hybrid approaches, combining the broad exploratory power of methods like Raygun with targeted directed evolution or function-specific experimental assays, would be critical for pushing the boundaries of protein engineering.

Looking forward, Raygun opens exciting possibilities for protein engineering. The novel approach of representing proteins as probability distributions may be useful in other areas of computational biology. Raygun's ability to efficiently generate diverse, functional protein variants while preserving core structural elements could accelerate drug discovery, particularly for biologic therapeutics, and advance synthetic biology applications such as biosensor design. The framework's flexibility in handling both substitutions and large-scale indels expands the universe of proteins derivable from a single template, potentially catalyzing new approaches to protein design.

Declarations

None whatsoever.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, Jun 2024.

[2] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X. Lu, Nicolo Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023.

[3] Alex Bateman, Lachlan Coin, Richard Durbin, Robert D Finn, Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik LL Sonnhammer, et al. The pfam protein families database. *Nucleic acids research*, 32(suppl_1):D138–D141, 2004.

[4] Suhaas Bhat, Kalyan Palepu, Lauren Hong, Joey Mao, Tianzheng Ye, Rema Iyer, Lin Zhao, Tianlai Chen, Sophia Vincoff, Rio Watson, Tian Wang, Divya Srijay, Venkata Srikar Kavirayuni, Kseniia Kholina, Shrey Goel, Pranay Vure, Aniruddha J Desphande, Scott H Soderling, Matthew P DeLisa, and Pranam Chatterjee. De novo design of peptide binders to conformationally diverse targets with contrastive language modeling. *bioRxivorg*, July 2024.

[5] Nadav Brandes, Grant Goldman, Charlotte H. Wang, Chun Jimmie Ye, and Vasilis Ntranos. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 55(9):1512–1522, Sep 2023.

[6] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 02 2022.

[7] Tomer Cohen and Dina Schneidman-Duhovny. Epitope-specific antibody design using diffusion models on the latent space of ESM embeddings. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

[8] A B Cubitt, R Heim, S R Adams, A E Boyd, L A Gross, and R Y Tsien. Understanding, improving and using green fluorescent proteins. *Trends Biochem. Sci.*, 20(11):448–455, November 1995.

[9] Sean R Eddy. Where did the blosum62 alignment score matrix come from? *Nature biotechnology*, 22(8):1035–1036, 2004.

[10] Robert D Finn, Jody Clements, and Sean R Eddy. Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37, 2011.

[11] Nathan C. Frey, Daniel Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, Andreas Loukas, Vladimir Gligorijevic, and Saeed Saremi. Protein discovery with discrete walk-jump sampling, 2024.

[12] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford university press, 2020.

[13] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024.

[14] Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, Feb 2024.

[15] Kazutaka Katoh and Daron M Standley. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, 30(4):772–780, April 2013.

[16] Talley J Lambert. Fpbase: a community-editable fluorescent protein database. *Nature methods*, 16(4):277–278, 2019.

[17] Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds, 2023.

[18] Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. 2024.

[19] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[20] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[21] Mark Lorch. 34C3Proteins: nature's nano-machines. In *Biochemistry: A Very Short Introduction*. Oxford University Press, 05 2021.

[22] Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, Aug 2023.

[23] Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

[24] Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[25] Suresh Pokharel, Pawel Pratyush, Michael Heinzinger, Robert H. Newman, and Dukka B. KC. Improving protein succinylation sites prediction using embeddings from protein language model. *Scientific Reports*, 12(1):16933, Oct 2022.

[26] Erik A Rodriguez, Robert E Campbell, John Y Lin, Michael Z Lin, Atsushi Miyawaki, Amy E Palmer, Xiaokun Shu, Jin Zhang, and Roger Y Tsien. The growing and glowing toolbox of fluorescent and photoactive proteins. *Trends Biochem. Sci.*, 42(2):111–129, February 2017.

[27] Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, Mikhail S. Baranov, Onuralp Soylemez, Natalya S. Bogatyreva, Peter K. Vlasov, Evgeny S. Egorov, Maria D. Logacheva, Alexey S. Kondrashov, Dmitry M. Chudakov, Ekaterina V. Putintseva, Ilgar Z. Mamedov, Dan S. Tawfik, Konstantin A. Lukyanov, and Fyodor A. Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, May 2016.

[28] Robert J Serfling. Contributions to central limit theory for dependent variables. *The Annals of Mathematical Statistics*, 39(4):1158–1175, 1968.

[29] Nathan C Shaner, George H Patterson, and Michael W Davidson. Advances in fluorescent protein technology. *Journal of cell science*, 120(24):4247–4260, 2007.

[30] Rohit Singh, Kapil Devkota, Samuel Sledzieski, Bonnie Berger, and Lenore Cowen. Topsy-turvy: integrating a global view into sequence-based ppi prediction. *Bioinformatics*, 38:i264–i272, 06 2022.

[31] Rohit Singh, Samuel Sledzieski, Bryan Bryson, Lenore Cowen, and Bonnie Berger. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proceedings of the National Academy of Sciences*, 120(24):e2220778120, 2023.

[32] Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. D-script translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982.e6, 2021.

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[34] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

[35] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, 2023.

[36] Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, Aug 2023.

[37] Carter J Wilson, Wing-Yiu Choy, and Mikko Karttunen. Alphafold2: a role for disordered protein/region prediction? *International Journal of Molecular Sciences*, 23(9):4591, 2022.

[38] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[39] Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

A Methods

A.1 Model architecture

The "T-Block" layers are arranged before and after the "Reduction" and "Repetition" layers to add model complexity preceding and succeeding the length-transforming operations. In the Raygun encoder, a T-Block output is usually the input to another T-Block and the Reduction layer. The T-Block pairs, that are before and after Reduction, are parameter-shared; we use 10 of them in total. All the reduced T-Block outputs in the encoder are aggregated in the final layer. There, the fixed-length representations are first concatenated together, and then projected to 1280 dimensional space (i.e. the embedding dimension of ESM-2).

Similarly in the Raygun decoder, the T-Blocks are also usually arranged in pairs before and after the Repetition operation, although their parameters are not shared. The target length provided by the user is used to project the fixed-dimensional embeddings back to the variable-length space. All the projected outputs coming from the Reduction layer, after passing through the T-Blocks, are then aggregated in the final projection layer. Similar to encoder, they are first concatenated together before projecting back to 1280 dimensional space. We use 21 T-Blocks in the Raygun decoder: 10 before the Repetition operation and 11 after. The overal Raygun architecture is described in **Figure** A.1.

We now proceed to describe the internal architecture of the Repetition and Reduction layers.

A.1.1 Reduction Layer

The Reduction block is a non-parametric layer that transfers variable-length embeddings to a fixed-length space and is vital in the fixed-length sampling process. Given an ESM-2 embedding $\boldsymbol{E} \in \mathbb{R}^{n \times 1280}$ (1280 being the ESM-2 650M dimension), the Reduction block returns two fixed-length embeddings: mean and standard deviation matrices, each of dimensions $\mathbb{R}^{K \times 1280}$, where K is fixed to 50 in our experiments.

Dividing the variable length embeddings into non-contiguous blocks is tricky if the sequence length n is not an exact multiple of the fixed-length dimension K. We addressed this by allowing an extra slack of size 1 to the blocks at the beginning and the end of the embedding. For example, suppose the sequence length is 543 and $K = 50$. We chunked the embedding from this sequence by allowing the first 22 and the last 21 blocks to have a size of 11, while fixing the middle 7 blocks to size 10 ($22 \times 11 + 7 \times 10 + 21 \times 11 = 543$).

We describe the Reduction pseudo-code in **Algorithm 1**

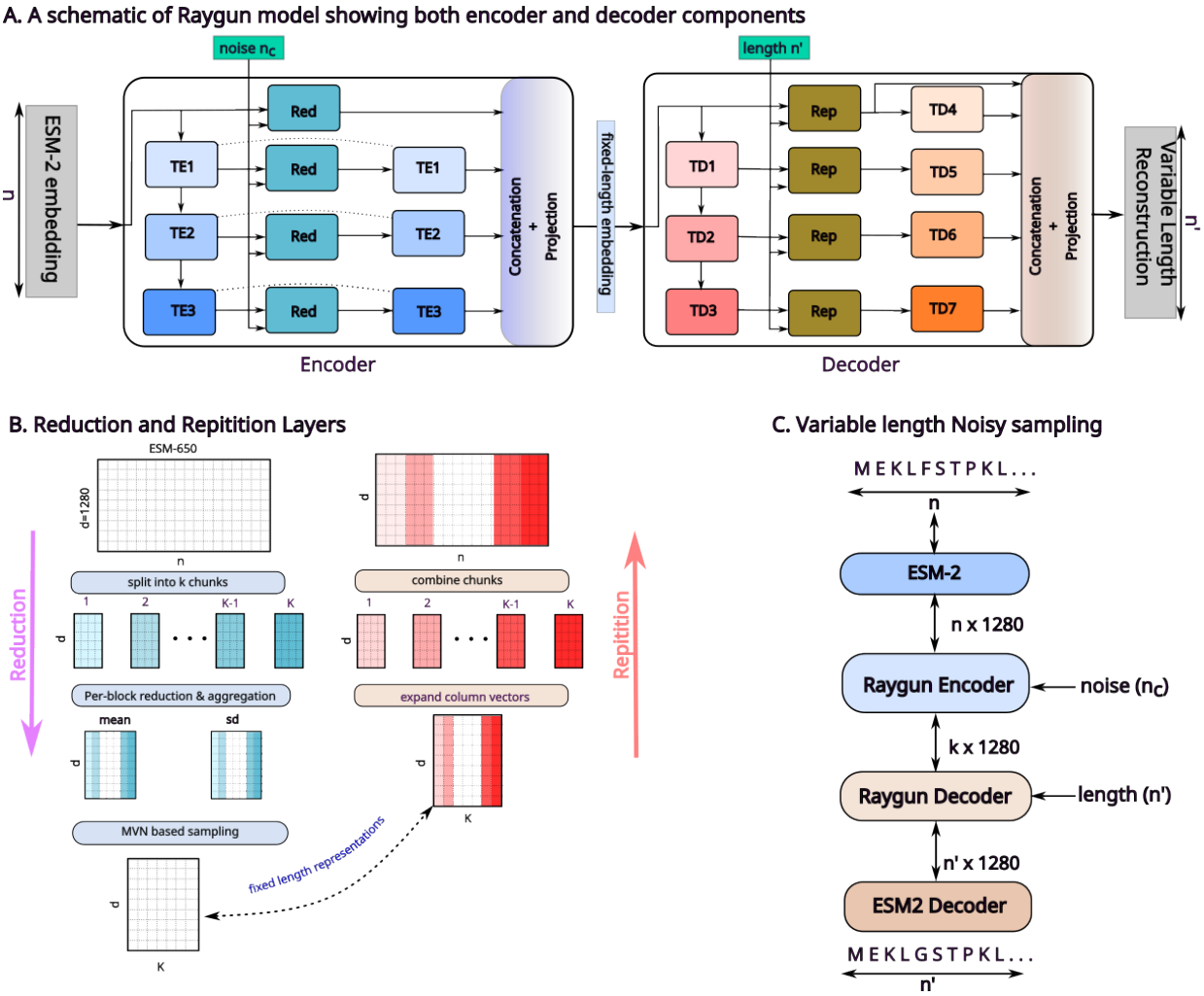

Figure A.1: Raygun architecture

A.1.2 Repetition Layer

The Repetition layer, which complements the Reduction layer, processes the fixed-length representations from the Raygun encoder and translates them into a variable-length space. This layer performs the length transformation by broadcasting each column vector of the fixed-dimensional representation, ensuring that the resulting combined embeddings have the desired length. Similar to the Reduction layer, the expansion process becomes more complex when the target length n is not an exact multiple of K. To address this, we allow a portion of the initial and final column vectors to be broadcasted with an additional value of 1. For example, if the target length is 543, the Repetition layer broadcasts the first 22 and the last 21 column vectors into matrices of size $\mathbb{R}^{111280}$. The remaining 7 column vectors in the middle are broadcasted into matrices of size $\mathbb{R}^{10\times1280}$.

The detailed algorithmic description of the Repetition layer is provided in **Algorithm 3**.

Algorithm 1 Forward operation of Raygun Reduction Layer

Input: esm-2 embedding $\boldsymbol{E} \in \mathbb{R}^{n \times 1280}$, noise factor n_c
Output: fixed length sample $\boldsymbol{E_{raygun}} \in \mathbb{R}^{K \times 1280}$
$mws = n//K; \quad gap = n - K \cdot mws; \quad mid = K - gap;$ ▷ mws = minimum window size
if $gap\%2 == 0$ **then**
  $gr = gl = gap/2$
else
  $gr = \lfloor gap/2 \rfloor; gl = gr + 1$
end if
$\boldsymbol{Es} = \boldsymbol{E}[0 : gl \cdot (mws + 1), :];$
$\boldsymbol{Em} = \boldsymbol{E}[gl \cdot (mws + 1) : gl \cdot (mws + 1) + mid \cdot mws, :]$
$\boldsymbol{Ee} = \boldsymbol{E}[n - gr \cdot (mws + 1) :, :];$
$\boldsymbol{Ms}, \boldsymbol{Ss} = GetMeanStd(\boldsymbol{Es}, mws + 1)$
$\boldsymbol{Mm}, \boldsymbol{Sm} = GetMeanStd(\boldsymbol{Em}, mws)$
$\boldsymbol{Me}, \boldsymbol{Se} = GetMeanStd(\boldsymbol{Es}, mws + 1)$
$\boldsymbol{M} = \text{Concatenate}(\boldsymbol{Ms}, \boldsymbol{Mm}, \boldsymbol{Me}, dim = 0)$
$\boldsymbol{S} = \text{Concatenate}(\boldsymbol{Ss}, \boldsymbol{Sm}, \boldsymbol{Se}, dim = 0)$
Sample $\boldsymbol{n_{raygun}} = Uniform(0, n_c), \boldsymbol{N} \sim \mathcal{N}(0, 1)$
return $\boldsymbol{E_{raygun}} = \boldsymbol{M} + \boldsymbol{n_{raygun}} \boldsymbol{N} \odot \boldsymbol{S}$

Algorithm 2 GetMeanStd()
!ht

Input: $r \in \mathbb{Z}^+, \boldsymbol{X} \in \mathbb{R}^{rk \times 1280}$
Output: $\boldsymbol{m}, \boldsymbol{s} \in \mathbb{R}^{k \times 1280}$
for $i \in 1, \ldots, r$ **do**
  $\boldsymbol{m}[i, :] = mean(\boldsymbol{X}[ik : (i + 1)k, :])$
  $\boldsymbol{s}[i, :] = sd(\boldsymbol{X}[ik : (i + 1)k, :])$
end for
return $\boldsymbol{m}, \boldsymbol{s}$

Algorithm 3 Forward operation of Raygun Repetition layer

Input: Fixed length representation $\boldsymbol{E_{raygun}} \in \mathbb{R}^{K \times 1280}$, $\boldsymbol{t_l} \in \mathbb{Z}^+$ ▷ $\boldsymbol{t_l}$ = target length
Output: Variable length representation $\boldsymbol{T_{raygun}} \in \mathbb{R}^{\boldsymbol{t_l} \times 1280}$
$mws = \boldsymbol{t_l}//K; \quad gap = \boldsymbol{t_l} - K \cdot mws; \quad mid = K - gap;$ ▷ mws = minimum window size
if $gap\%2 == 0$ **then**
  $gr = gl = gap/2$
else
  $gr = \lfloor gap/2 \rfloor; gl = gr + 1$
end if
$\boldsymbol{T_s} = Fill(\boldsymbol{E_{raygun}}[: gl, :], mws + 1)$
$\boldsymbol{T_m} = Fill(\boldsymbol{E_{raygun}}[gl : gl + mid, :], mws)$
$\boldsymbol{T_e} = Fill(\boldsymbol{E_{raygun}}[gl + mid :, :], mws + 1)$
return $\boldsymbol{T_{raygun}} = Concat(\boldsymbol{T_s}, \boldsymbol{T_m}, \boldsymbol{T_e}, dim = 0)$

A.1.3 T-Block layer

T-Block layers are deep neural network modules trained to optimize embeddings by enhancing latent local and global properties. Each T-Block layer comprises an ESM Transformer for global properties and a 1D-convolution block for local relationships. The internal PyTorch architecture of each T-Block is

Algorithm 4 Fill()

Input: $r \in \mathbb{Z}^+, \boldsymbol{X} \in \mathbb{R}^{k \times 1280}$
Output: $\boldsymbol{Y} \in \mathbb{R}^{kr \times 1280}$
Initialize $\boldsymbol{Y}$
for $i \in 1, \ldots, k$ **do**
  $\boldsymbol{Y}[ir : (i+1)r, :] = broadcast(X[i, :], r)$
end for
return $\boldsymbol{Y}$

```
Block(
  (encoder): TransformerLayer(
    (self_attn): MultiheadAttention(
      (k_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (v_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (q_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (out_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (rot_emb): RotaryEmbedding()
    )
    (self_attn_layer_norm): ESM1LayerNorm()
    (fc1): Linear(in_features=1280, out_features=2560, bias=True)
    (fc2): Linear(in_features=2560, out_features=1280, bias=True)
    (final_layer_norm): ESM1LayerNorm()
  )
  (convblock): Sequential(
    (0): Rearrange('b n c -> b c n')
    (1): Conv1d(1280, 640, kernel_size=(7,), stride=(1,), padding=same)
    (2): SiLU()
    (3): Conv1d(640, 320, kernel_size=(3,), stride=(1,), padding=same)
    (4): SiLU()
    (5): Conv1d(320, 640, kernel_size=(7,), stride=(1,), padding=same)
    (6): Rearrange('b c n -> b n c')
  )
  (final): Linear(in_features=640, out_features=1280, bias=True)
)
```

Figure A.2: The internal architecture of T-block of the pretrained Raygun model

shown in **Figure A.2**.

A.2 Training Raygun and optimization losses

The Raygun encoder and decoder components are jointly trained to ensure that the decoder can accurately reconstruct the variable-length space from the fixed-length encoding produced by the encoder. To achieve this, we sampled 94,734 proteins from the UniRef50 dataset, carefully maintaining length diversity in the selection. During training, the model takes a protein of length n, applies the Raygun encoder to project it into a fixed-length representation where $K = 50$, and then reconstructs the embedding back to its original length n using the Raygun decoder. The output from the Raygun decoder is further converted to amino acid logits using a pre-trained ESM-to-tokens decoder. Backpropagation is

then performed to minimize the following losses:

1. Cross-Entropy Loss (L_{ce}): to ensure that the predicted tokens match the input tokens.
2. Reconstruction Loss (L_{rr}): to ensure that the reconstructed ESM embedding matches the input.
3. Replicate Loss (L_{rp}): Let a protein p has a length n and suppose we generated a new protein p' of length n' using Raygun. Then, this loss is designed to ensure that the fixed-length embedding of p is close to that of p'

Let p be the input protein of length n, p' the Reconstructed Raygun sequence, ESM_p the ESM-650M embedding of p, $RGUN_p^{(50)}$ be the fixed length encoding of p and $RGUN_p^n$ be the reconstruction of p using Raygun to length n. Let p'' be another Raygun sequence obtained from p of length $n' < n$. Then the total and constituent losses become:

$$L_{total} = L_{ce} + L_{rr} + L_{rp} \tag{1}$$

$$L_{ce} = CrossEntropy(p, p') \tag{2}$$

$$L_{rr} = \|ESM_p - RGUN_p^n\|_2 \tag{3}$$

$$L_{rp} = \|RGUN_p^{(50)} - RGUN_{p''}^{(50)}\|_2 \tag{4}$$

A.2.1 Blosum score

We used a Blosum-based sequence identity score, or simply "Blosum score" to evaluate the accuracy of Raygun on the validation dataset. Given the template protein S and the predicted Raygun candidate S', this score is computed as the ratio:

$$Blosum\ score(S, S') = \frac{\sum_i blosum62(S(i), S'(i))}{\sum_i blosum62(S(i), S(i))} \tag{5}$$

Unlike sequence identity (seq. id), Blosum score can range between -1 and 1. Additionally, having a good Blosum score is a stronger condition than having a seq. id score of a similar magnitude, as the blosum62 score of two amino acids is usually negative when they do not match.

A.2.2 Choice of the reduction parameter K

As a preliminary experiment, we aimed to determine the optimal fixed-length reduction parameter K for Raygun. To this end, we randomly selected 103,463 sequences from SwissProt, filtered at 50% sequence identity, and split them into 90,524 training and 12,939 validation sequences. We then generated the corresponding Blosum scores for $K = 25$ and $K = 50$ after running the models for 3 epochs. Our results showed that $K = 50$ consistently outperformed $K = 25$ in both the training and validation phases

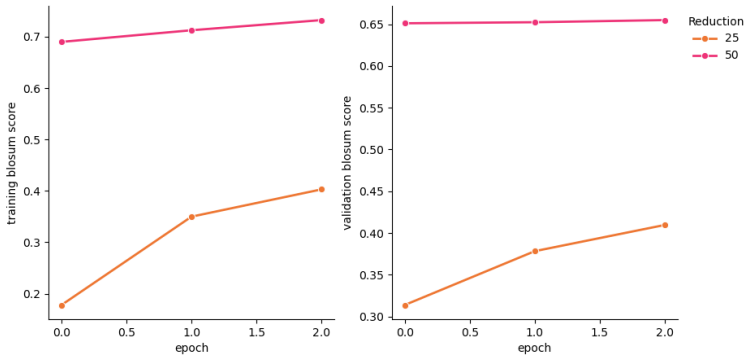

Figure A.3: Training and validation Blosum scores for $K = 25, 50$ on the Swissprot dataset

(**Figure A.3**). Consequently, we chose $K = 50$ as the default size for the fixed-length representation.

A.2.3 Training the model for $K = 50$

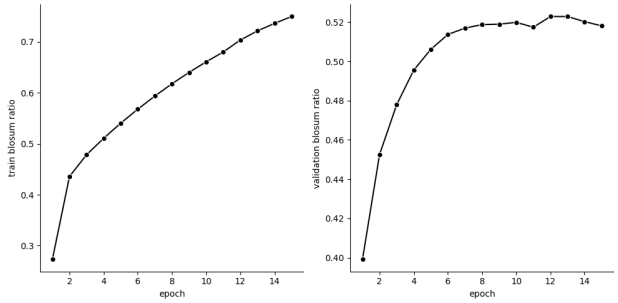

Figure A.4: Train and validation Blosum scores for epochs1-15

We trained our Raygun model for 15 epochs on 6 A100 GPUs using a Distributed Data Parallel (DDP) approach. Each epoch took approximately 12 hours (8 days in total). Highest validation Blosum score of 0.51 was reported on epoch 12. We show the train and test Blosum scores in **Figure A.4**.

A.3 Filtering Raygun samples

The generation speed of Raygun gives us enough flexibility to apply many filtering approaches to improve the quality of the generated candidates. We discuss some of these approaches below

A.3.1 pseudo-Log likelihood and repeats penalization

As a first filtering step, we used pseudo-log likelihood [5] to filter the Raygun samples. Given a candidate sequence, pLL uses the esm-2 generated logits to compute the degree to which esm-2 identifies the sequence as a viable protein, rather than an arbitrary sequence.

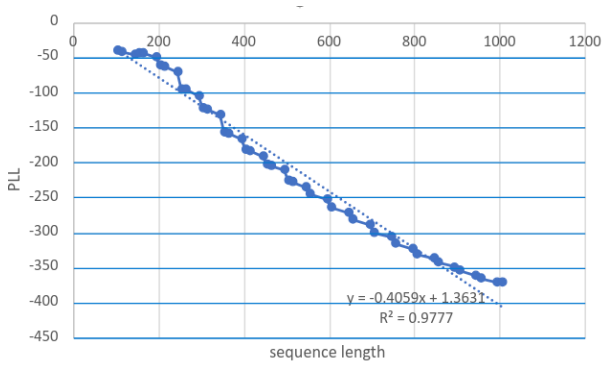

Figure A.5: Plot of sequence length vs pLL scores observed for a sample of Uniref50 proteins

One challenge with using pLL out of the box is that it is length-dependent and more suited for evaluating substitutions rather than indels. In our generation pipelines, we typically explored multiple output length ranges. Within a particular range, we needed to adjust pLL for minor length variations, which we did empirically. We observed an approximately linear relationship between sequence length and the pLL score (Figure A.5).

To adjust for length variations, we updated the pLL score to make it length-invariant. The updated $pLL_{invar}(S)$, where S is the input sequence, becomes

$$pLL_{invar}(S) = \frac{pLL(S)}{|-0.406 \cdot len(S) + 1.363|} \tag{6}$$

Additionally, in order to discourage sequences with long repeats, we introduced a small sequence-repeat penalty. A similar strategy has been used in other generation methods [2]. We measured the length of sub-sequences with more than 3 repeats $R_{rep>3}$ with the total sequence length. We then update pLL_{invar} again using $R_{rep>3}$ in the following way:

$$pLL_{invar}(S) \leftarrow (0.5 + R_{rep>3}(S)) \cdot pLL_{invar}(S) \tag{7}$$

This updated metric, which is always negative, is then used to find candidate sequences with higher fitness (i.e. having pLL_{invar} values closer to 0).

A.3.2 Using HMMER for evaluating PFAM domains

In PFAM-based experiments (Section 2.3 of the main paper), we used HMMER [10] as an evaluation tool to determine the percentage of generated Raygun samples that retain the original PFAM domain of the template. After pLL filtering, the Raygun candidates for each PFAM domain were processed through

```
FluorescentHead(
  (encoder): TransformerLayer(
    (self_attn): MultiheadAttention(
      (k_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (v_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (q_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (out_proj): Linear(in_features=1280, out_features=1280, bias=True)
      (rot_emb): RotaryEmbedding()
    )
    (self_attn_layer_norm): ESM1LayerNorm()
    (fc1): Linear(in_features=1280, out_features=2560, bias=True)
    (fc2): Linear(in_features=2560, out_features=1280, bias=True)
    (final_layer_norm): ESM1LayerNorm()
  )
  (inter): Sequential(
    (0): Linear(in_features=1280, out_features=320, bias=True)
    (1): Dropout(p=0.2, inplace=False)
    (2): ReLU()
    (3): Linear(in_features=320, out_features=32, bias=True)
    (4): ReLU()
  )
  (final): Linear(in_features=32, out_features=1, bias=True)
)
```

Figure A.6: Internal architecture of the Fluorescent head

HMMER. The ratio of candidates that retained the domain after length modifications was then used to evaluate Raygun's ability to preserve domain information in its generated samples.

A.3.3 Using HMMER as a filtering tool for fluorescent protein generation

In the fluorescent protein experiments, we used HMMER as an additional filtering tool to discard Raygun-generated candidates lacking the appropriate PFAM domain. For new FP generation, we specifically checked if the candidates belonged to the PFAM domain "PF01353". The PFAM score threshold was set to 20 during filtering.

A.3.4 Using known GFP brightness information for additional FP filtering

In addition to *pLL* and HMMER-based filtering, we also used existing brightness data obtained from a deep mutational scan (DMS) assay on avGFP [27] to train a deep model and select candidates predicted to be the brightest. The DMS dataset contained brightness results for 54,025 sequences. We constructed a simple fluorescent head on top of ESM-2 (with its weights frozen) and trained it to predict the brightness values of input DMS variants of avGFP. The internal architecture of the fluorescent head is shown in Figure A.6. We trained this model for 50 epochs.

When deploying this trained brightness model, we aimed to ensure that it would not negatively impact sequence diversity by selecting a narrow range of sequence diversity. Therefore, we first clustered the candidates filtered by pLL and HMMER into 15 disjoint clusters. We then applied the brightness model to each of the 15 clusters and obtained the 15 brightest predicted sequences for both eGFP and mCherry. The clustering was done as follows:

1. For the remaining candidates, perform Multiple Sequence Alignment (MSA) using MAFFT [15].
2. Compute pairwise Blosum scores between all candidate pairs aligned in the MSA. Use these pairwise similarity scores to construct an affinity matrix.
3. Finally, use spectral clustering to group the candidates into 15 non-overlapping clusters.

In the final stage, we generated AlphaFold3 structures for these 30 candidates in total and used the obtained PDBs to compute TM-score and pLDDT. We used this structural information to manually select 4 candidates for each FP, ensuring that candidates across diverse lengths were included. The amino acid sequences of these 8 candidates are provided in **Figure A.7**.

A.3.5 Advantages of Recycling

Although Raygun can be used to perform one-shot generation, we hypothesized that the diversity of the generated candidates would be greater if we applied a one-step recycling procedure: taking Raygun output and passing it back again to the encoder, in order to get a recycled product. Recycling has been shown to be a powerful tool to enhance the reach of a model without adding additional model complexity, and has been widely used in methods like AlphaFold2 and ESMFold. To test this hypothesis, we took 5 proteins from the Rhodopsin family (PF00001): *ADA1D_HUMAN*, *CCR1_HUMAN*, *FSHR_BOVIN*, *LPAR6_CHICK* and *NTR_HUMAN*, as templates, fine-tuned Raygun on these 5 proteins and used them to generate 38 candidates per template, spread over the length ranges 90 to 1010. We set the Raygun noise-factor to 0.5, and used pLL as a filtering step. We then used HMMER to estimate the percentage of these candidates that retained the "PF00001" domain.

We performed this experiment for the "no-cycle" and "1-step recycle" settings, and found that the number of candidates with the PFAM domain retained after recycling was significantly higher at 68/190 than the candidates generated without doing any recycling (50/190). Therefore, the default invocation of Raygun for protein generation includes 1-step recycling, which can be also be disabled by the user.

```
>xfp01
MVSKGEELFTGVVPILVELDGDVNGHKFSVSGEGSGDATYGKLTLKFINTTGLPVPVTLV
TLTYVQCSRYDDKMQDFFKSCPEGVQERTFFKDDGNYKTRAEVKFEGDTLVNRIELKGID
FKEDGNILGHKLEYNYNHNVYMADKKNGKVNMIRHNEDGVQLDDHQQTPIGGGVLLPDNH
YLSTQSALSKDPNEKRDHMVLLEFVTAAGITLGTDELYK
>xfp02
MVSKGEELFTGVVPILVELDGDVNGHKFSVSGEGEDATYGKTLKFITTKLPVPPTLVTLY
GVQFSRYPDMKHHFFKSAPEGVQERIFFKDDGNYKTRAEVKFEGDTLVNRIELKGIDFKE
DGNHLGHKLEYNHSHNVIMAKKQNGVVNFIRHNEDGVQLADYQQNPIGDGVLLDNHYSTQ
SASKDPNEKRDHMVLLEFVTAAGITLGMDELYK
>xfp03
MVSKGEELFTVVVPILVELDGDVNGHKFSVSGEGEGDDTYGKLTLKFICTTGKLPVPPPT
LVTLYGVQCCKYPDMKQDFFKASPEGVKERIFFKDDGNYKTRAEVKFEGDTLVNRIELKG
IDFKEDGNILGHKLEYNHSHNVLIADDKNGKCNFFRHNEDGVQLADHYQQNTPIGDGGVL
LPDNHYLSTQSAASKDPNEKRDHMVLLEFVTAAGITHGMDELYK
>xfp04
MVSKGEELFTVPVPKLVELDGSVNGHKFSVSGEGEGDATYGKLTLKFICTTGKLPVPWPT
LVTTLTYGVQCCSRYPDHMKQHDFFKKMPEGVQRTIFFKDDGNYKTVAEVKFEGDTLVNR
IELKGIDFKEDPNIVGHCLEYNYHHNVIMADKKNGIKVHFKDRQNIEDGSVSLADHYQQN
TPIGDGPHLLPDNHYLSTQSALNKDPNEKRDHMVLLEFVTAAGITLGMDELYK
>xfp05
MVSKGEEDNMAIIKEFMRKVHEGSVGNEFIEGEGGRPEGTQTKLKVKGGLPFADILSQFM
YGKAYKHPDDPDYKLSFPEGFKWERVMNFEDGGVVTVTQDSSLQDGEFIYKVKLRGTNFP
SDGPVMQKKTMGWASSRRYPEGALGEIKQLKLKGGHYAEVKTYKKKPVQLGAYNNIKDIT
SHEDYIVEQERAEGRHSTGGMDELYK
>xfp06
MVSKGEEDNMAIIKEFMRFKVHMEGSVNGHEFEIEGEGEGRPYEGTQTAKLKTKGGLPFA
DILSQFMYGKAYKHPDPPDYKLSFPEGFKWERVMNFEDGGVVTVTQDSSLQDGEFIYKVK
LRGTNFPSDGPVMQKKTMGWASSRRYPEGALKEIKRLKLDGGHYAEVKTYKAKPVQLPGA
YNVNIKLDITSHNEDYTIVEQYERAEGRHSTGGMDELYK
>xfp07
MVSKGEEDNMAIIKEFMRFKVHMEGSVNGHEFEIEGEGEGRPYEGTQTAKLKLTKGGPLP
FAWDILSPQFMYGSKAYVKHPADIPDLKLSFPEGFKWERVMNFEDGGVVTVTQDSSLQDG
EFIYKVKLRGTNFPSDGPVMQKKTMGWASSSRRYDEDGALKGEIKQRLKLKDGGHYDAEV
KTTYKAKKPVQLPGAYNVNIKLDITSHNEDDTIVEQYERAEGRHSTGGMDELYK
>xfp08
MVKGEDNMAIKEFMFKVHEGSVGHEFEEGEGGRPEGTSAKLKVKGGPPFADILSQFMYSK
AYKHPDDPDLKLSFPEGFKWERVMNFEDGGVVTVTQDSSLQDEFIYKVKLRGTNFPSDGP
VMQKKTMGWASSERYPEGALKEIKRLKLKGGHYAEVKTYKKKPVQPGAYNNIKDITSHED
YIVEQERAEGHSTGGDEYK
```

Figure A.7: Amino acid sequences of the 8 Raygun-generated XFP candidates

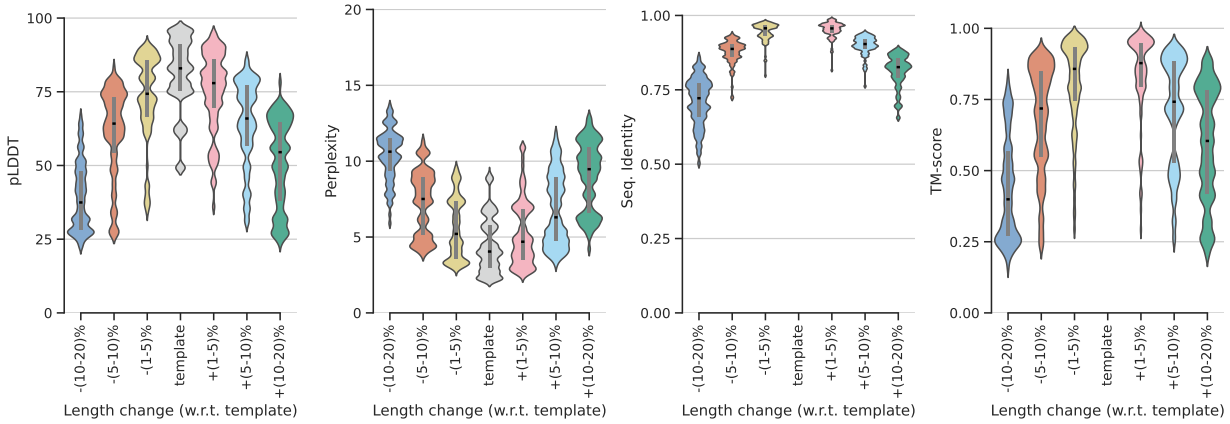

Figure A.8: pLDDT, perplexity, seq. identity and TM-score results, after changing candidates lengths by different margins, for noise-factor 0.5

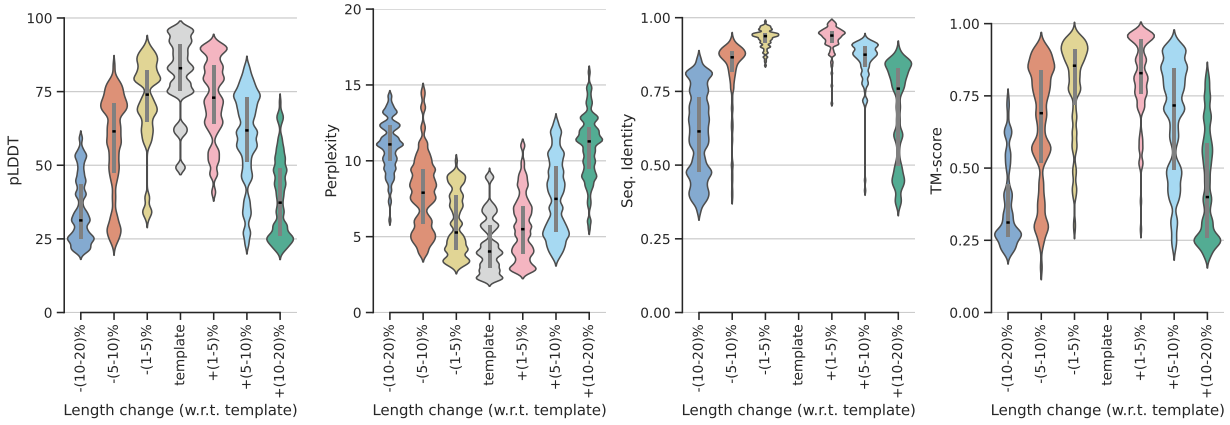

Figure A.9: pLDDT, perplexity, seq. identity and TM-score results, after changing candidates lengths by different margins, for noise-factor 1.0